Some Results on the Matrix Multiplication Problem

L. H. Harper
University of California at Riverside

J. E. Savage
Brown University

Three results on the multiplication of two $n \times n$ matrices are presented. They contribute to our understanding of the complexity of matrix multiplication, and so of code decoding, tracking accuracy computation, antenna structural analysis and other DSN computational tasks.

I. Introduction

Recent results (Refs. 1 and 2) have shown that two $n \times n$ matrices can be multiplied with a number of arithmetic operations which grows as n^a, where $a = \log_2 7 \simeq 2.81$. For large n, this represents a significant reduction from the number of operations which are performed when the defining equations for matrix multiplication are used.

In this article we contribute to the matrix multiplication problem in the following ways: (1) conditions are given under which the conventional method for multiplying two matrices is optimal, (2) an n^2 lower bound on the number of arithmetic operations necessary for the multiplication of two $n \times n$ matrices is derived, and (3) a nearly optimum algorithm for the computation of any one (but arbitrary) element of a product for two matrices is presented.

II. Fan-Out 1 Complexity of Matrix Multiplication

In another article in this volume,[1] "straight-line" algorithms are defined. When restricted to arithmetic operations, these algorithms make repeated use of addition, subtraction, multiplication, and division over the reals. Straight-line algorithms have only these operations, and no loops or conditional branches are permitted. An algorithm is said to compute functions $f_1, \cdots, f_L$ if these

[1] Johnson, D., Savage, J., and Welch, L., "Combinational Complexity Measures as a Function of Fan-Out."

functions are computed at some of the steps of the algorithm. An algorithm is also said to have fan-out s if no computation by the algorithm is used externally or internally more than s times. Then, the *s fan-out combinational complexity* of $f_1, \cdots, f_L$, $C_s(f_1, \cdots, f_L)$ is the smallest number of computation steps in any straight-line algorithm of fan-out s which computes $f_1, \cdots, f_L$.

Let $A = \{a_{ij}\}$, $B = \{b_{ij}\}$ be two $n \times n$ matrices over the reals. Let $D = \{d_{ij}\}$ be the result of multiplying A and B. Then $D = AB$ and

$$d_{ij} = \sum_{k=1}^{n} a_{ik} b_{kj}$$

Theorem. *Let $C_1(D)$ be the 1 fan-out combinational complexity of the n^2 functions of $D = AB$, where A, B are $n \times n$ arbitrary matrices over the reals. Then, $C_1(D) =$* $n^2(2n-1)$.

Proof. The standard method for computing D uses n^3 multiplications and $n^2(n-1)$ additions so that

$$C_1(D) \leqq n^2(2n-1)$$

Let β be an optimal straight-line algorithm with fan-out 1 which computes D. Then, the computations used to compute any two elements of D must be different since the algorithm uses the result of a computation only once. Thus, the complexity of D is the sum of complexities of the functions d_{ij}, $1 \leqq i, j \leqq n$. But each of these depends on $2n$ variables and it can be shown[2] that any function dependent on N variable requires at least $N-1$ binary computations. Therefore, we have that $C_1(D) \geqq n^2(2n-1)$ which is exactly the upper bound.

Q.E.D.

III. A Lower Bound on the Combinational Complexity of Matrix Multiplication

In this section we develop a lower bound to $C_s(D)$, $s \geqq 2$, by lower bounding the complexity of the trace of D, $\operatorname{tr}(D)$. We observe that

$$\operatorname{tr}(D) = d_{11} + d_{22} + \cdots + d_{nn}$$

and

$$C_s(\operatorname{tr}(D)) \leqq C_s(d_{11}, d_{22}, \cdots, d_{nn}) + C_s(S(x_1, x_L, \cdots, x_n))$$

where $S(x_1, x_2, \cdots, x_n)$ is the sum of $x_1, x_2, \cdots, x_n$. Since $C_s(S(x_1, x_2, \cdots, x_n)) \leqq n-1$, we have

$$C_s(D) \geqq C_s(d_{11}, d_{22}, \cdots, d_{nn}) \geqq C_s(\operatorname{tr}(D)) - (n-1)$$

However, the function $\operatorname{tr}(D)$ depends on the $2n^2$ variable entries of A and B so, $C_s(\operatorname{tr}(D)) \geqq 2n^2 - 1$. We conclude that:

Theorem. $C_s(D) \geqq 2n^2 - n + 1, \qquad s \geqq 2$

It has been conjectured (Ref. 3) that $C_s(D)$ must grow as n^2. If so, this bound establishes that this rate of growth cannot be improved.

IV. An Algorithm for Computing the Elements of a Matrix Product

Consider the function $f(i, j, A, B) = d_{ij}$ where $D = \{d_{ij}\} = AB$. This function computes an arbitrary element of AB and we shall show that $C_s(f)$, $s \geqq L$, grows as n^2 and that this rate of growth can be achieved.

Without excessive loss of generality, we restrict attention to matrices with binary elements and to addition and multiplication modulo 2. Then the integers i and j in $f(i, j, A, B)$ must be given a binary encoding. It is easily shown that f depends on all $2n^2$ entries in A and B so that, independently, of the encoding for i and j, $C_s(f) \geqq 2n^2 - 1$.

The following algorithm realizes f with $4n^2 - 1$ computations represent i by $(\alpha_1, \alpha_2, \cdots, \alpha_n)$ and j by $(\beta_1, \beta_2, \cdots, \beta_n)$ where

$$\alpha_e = \begin{cases} 1, & e = i \\ 0, & e \neq i \end{cases}$$

$$\beta_e = \begin{cases} 1, & e = j \\ 0, & e \neq j \end{cases}$$

Then,

$$\begin{aligned} f(i, j, A, B) &= \sum_{m=1}^{n} \left(\sum_{e=1}^{n} \alpha_e a_{em} \right) \left(\sum_{e'=1}^{n} \beta_{e'} b_{me'} \right) \\ &= d_{ij} \end{aligned}$$

and this algorithm has $(2n+1)n$ multiplications and $(2n+1)(n-1)$ additions for a total of $4n^2 - 1$ operations.

It is important to note that the function $f(i, j, A, B)$ pro-

[2] See Harper, L. H., and Savage, J. E., "Contributions to a Mathematical Theory of Complexity" (this volume).

duces an arbitrary element of AB. If some particular element of AB, say d_{12}, is to be computed, the representation

$$d_{12} = \sum_{k=1}^{n} a_{1k} b_{k2}$$

can be used to compute it with $2n - 1$ operations. It is the flexibility implicit in the definition of f that causes its complexity to grow as n^2.

V. Conclusion

Several contributions to the matrix multiplication problem have been given. It still remains to show whether Strassen's algorithm which requires $n^{\log_2 7}$ operations can be improved upon or not.

References

1. Strassen, V., "Gaussian Elimination is Not Optimal," *Numer. Math.*, Vol. 13, pp. 354–356, 1969.

2. Hopcroft, J. E., and Kerr, L. R., "On Minimizing the Number of Multiplications Necessary for Matrix Multiplication," *SIAM J. Appl. Math.*, Vol. 20, No. 1, pp. 35–36, Jan. 1971.

3. Fiduccia, C. L., "Fast Matrix Multiplication," in *Proceedings of the Third ACM Symposium on the Theory of Computing*, Shaker Heights, Ill., May 1971.